Low-Complexity Scheduling for Delay Minimization in D2D Communications using Network Coding

Mohammed S. Al-Abiad, Student Member, Md. Jahangir Hossain, Senior Member, IEEE, and Ahmed Douik, Member, IEEE

***Abstract*—In this letter, we consider the decoding delay minimizing problem for delivering a frame of packets to a set of user-devices (UDs) using instantly decodable network coding (IDNC). In the considered device-to-device (D2D) network, UDs have limited coverage zones that represent clusters and can speed up the delivery of the requested packets of other UDs by sending IDNC packets. The decoding delay minimization problem is a joint optimization problem of selecting the transmitting UDs and their coding decisions. In this work, we propose a low complexity, yet optimal, solution to the decoding delay minimization problem using graph pruning method. Our proposed innovative method introduces a sequential pruning algorithm that judiciously generates clusters that are certainly contributing to the network while simultaneously designing a new multi-layer IDNC graph. We also prove that the optimal solution to the problem can be achieved by the generated clusters by our proposed algorithm. Numerical results reveal that the proposed solution significantly reduces the computational complexity compared to the existing method with similar decoding delay performance.**

***Index Terms*—Device-to-device networks, clustering, IDNC graph, maximum weight clique.**

I. Introduction

THE popularity of smart user-devices (UDs) and the need to use data rate hungry applications such as video streaming and online gaming increase the mobile data traffic dramatically [1]. This proliferation of UDs and their demands impose a huge burden on the wireless networks and make efficient data delivery a key concern. Device-to-device (D2D) communication is a promising solution to support crucial applications, such as on-demand data delivery and distribution between spatially distributed UDs.

IDNC, a subclass of NC, is a promising technique for significantly minimizing delay over wireless erasure channels [4], [5]. IDNC suits most real-time applications due to its instantaneous and progressive decoding of packets. Besides, the simple process of encoding and decoding packets using XOR binary operation inscribes IDNC in different systems, e.g., video on-demand streaming [2]–[9]. Therefore, IDNC suits battery-limited D2D UDs [2], [3]. In IDNC, the decoding delay performance is an important metric as the decoding delay quantifies the ability of the transmitting UDs to generate innovative IDNC packets for a set of receiving UDs [5]. It increases by one unit for each UD that still requests packets and successfully receives a non-useful transmission [7]–[9]. In this work, we consider the decoding delay minimization problem in D2D network using IDNC.

The optimal IDNC schedule to the decoding delay minimization problem is shown to be computationally prohibitive. This is becasue the optimal IDNC schedule depends on all future transmissions and channel erasure probabilities for the whole transmission process until all UDs receive all their requested packets [5], [7]. In order to reduce the complexity, the widely adopted approach [5], [7] is to minimize the decoding delay in each transmission. In conventional point-to-multipoint (PMP) systems, solving the problem needs only an optimization over the packet combination selection. In IDNC-enabled D2D networks, UDs have limited transmission ranges (i.e., limited coverage zones that represent clusters) and already received some packets and missed some other packets. By exploiting such received and missed packets, UDs can cooperate to speed up the delivery of the missing packets of other UDs by sending IDNC packets. Therefore, IDNC-enabled partially-connected D2D network needs a careful optimization to select the transmitting UDs, their corresponding packet combinations, and D2D link erasures.

In [9], the authors considered the decoding delay minimization problem in IDNC-enabled D2D networks and showed that the optimal approach provides significant performance gain as compared to existing schemes. However, its complexity increases significantly with the number of generated clusters that is equal to the total number of UD combinations. Notably, the solution requires generating all the possible clusters whose union gives the optimal clustering approach. In this work, we develop a low-complexity, yet optimal, solution that involves the generation of only potential clusters. As such, the decoding delay in IDNC-enabled D2D networks is minimized. In particular, we derive the conditions under which clusters are beneficial for minimizing the decoding delay. Afterward, we develop a low-complexity graph pruning algorithm that judiciously generates clusters that are important for minimizing the decoding delay. Using these generated clusters, our proposed algorithm designs a new *multi-layer IDNC graph*, referred to ML-IDNC graph, where each vertex represents a cluster. We also prove that the optimal solution to the decoding delay minimization problem can be achieved by the generated clusters by our proposed algorithm. Numerical results show that the proposed solution offers similar performance of the optimal method in [9] with a significant reduction in the computational complexity.

II. System Model and Problem Formulation

A. System Model and Parameters

Consider a device-to-device (D2D) network consisting of N user-devices (UDs), denoted by the set $\mathcal{N} = \{1, \cdots, N\}$. We consider that UDs are interested in receiving a frame $\mathcal{M} = \{1, \cdots, M\}$ of M packets. We assume that UDs have received some packets in $\mathcal{M}$ from previous transmissions and requests a set of packets from the frame $\mathcal{M}$[1]. As assumed in [7]–[9], we consider that each packet in the frame $\mathcal{M}$ is received by at

Mohammed S. Al-Abiad and Md. Jahangir Hossain are with the School of Engineering, University of British Columbia, Kelowna, BC V1V 1V7, Canada (e-mail: m.saif@alumni.ubc.ca, jahangir.hossain@ubc.ca).

Ahmed Douik is with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: ahmed.douik@caltech.edu).

[1]The base station (BS) broadcasts the $\mathcal{M}$ packets to all the UDs and due to the channel impairments some of these packets are lost at the UDs as detailed in [7]–[9].

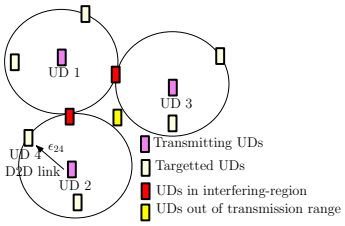

Fig. 1: A simple partially connected D2D network with 12 UDs. For simplicity, we plot the coverage zones of UDs 1, 2, and 3.

least one UD. The received and requested packets in $\mathcal{M}$ can be represented by the side information of the i-th UD as follows: 1) the *Has* set $\mathcal{H}_i$: previously received packets by UD i, 2) the *Wants* set $\mathcal{W}_i$: packets missed at UD i.

At each transmission slot, each UD can be a receiver or a transmitter that can transmit its acquired packets to other UDs via D2D links. Consequently, UDs cooperate among them to ensure that each UD in the network successfully receives all the $\mathcal{M}$ packets with the minimum possible decoding delay. We consider a realistic partially connected D2D network, where each UD can only target the subset of UDs in its coverage zone, denoted by $\mathcal{Z}_i$ of UD i and expressed as $\mathcal{Z}_i = \{j \in \mathcal{N} | z_{ij} = 1\}$. Let ϵ_{ij} be the packet erasure probability from UD i to UD j. These erasure probabilities $\{\epsilon_{ij}\}_{i,j\in\mathcal{N}}$ are assumed to be fixed and known to the BS. Accordingly, the transmitting UDs and packet combinations are selected based on the available information, such as the erasure patterns of the D2D links and the diversity of received and requested packets. We consider the standard assumption used in [8]–[10] that UDs use the same frequency band and transmit packets simultaneously. Hence, UDs located at the intersection of the transmission range of multiple transmitting UDs experience collision and no packets can be decoded. To avoid such collisions, UDs are grouped in disjoint clusters. For each transmitting UD, there is a cluster containing all UDs in its coverage zone. The system model is drawn in Fig. 1 that shows a partially connected D2D network.

The transmitting UDs and their packet combinations are selected to minimize the decoding delay that is defined as follows [8]–[10].

Definition 1*: At any transmission slot, UD i, with non-empty Wants set, experiences one unit increase of decoding delay if it does not receive any of its requested packets.*

B. Problem Formulation and Local IDNC Graph Construction

The decoding delay minimization problem over the whole transmission process is intractable [7]. Therefore, we consider the commonly and widely adopted approach that minimizes the decoding delay at each transmission slot.

Let $\mathcal{K} \in \mathcal{P}(\mathcal{N})$ denote the set of transmitting UDs and $\mathcal{T}$ denote the set of UDs that can receive multiple transmissions from the transmitting UDs, which can be expressed as $\mathcal{T} = \{i \notin \mathcal{K} | \exists (m,n) \in \mathcal{K}^2, m \neq n, i \in \mathcal{Z}_n \cap \mathcal{Z}_m\}$. Let $\mathcal{S}$ denote the set of UDs that are not in the transmission range of any transmitting UD, which can be expressed as $\mathcal{S} = \{i \in \mathcal{N} | \nexists j \in \mathcal{K}, i \in \mathcal{Z}_j\}$. Let $\mathcal{N}_w$ denote the set of UDs having non-empty *Wants* set. Now, let $\mathtt{p}_i \in \mathcal{P}(\mathcal{H}_i)$ denote the set of packets recombined by UD $i \in \mathcal{K}$ to be transmitted to the set of scheduled UDs $\mathtt{u}_i(\mathtt{p}_i)$, where $\mathcal{P}(\mathcal{H}_i)$ is the power set of $\mathcal{H}_i$. Therefore, the set of scheduled UDs $\mathtt{u}_i(\mathtt{p}_i)$ receive an instantly decodable transmission from UD i if: (i) the transmission contains one of the requested packets by UD j and (ii) UD j is in the coverage zone of UD i.

The problem of minimizing the decoding delay increase is a joint optimization problem of selecting the set of transmitting UDs and their packet combinations. Let $D(\mathcal{K}, \mathtt{p}(\mathcal{K}))$ denote the total decoding delay increase. By Definition 1 and the aforementioned system configuration, the decoding delay increase $d_j(\mathcal{K}, \mathtt{p}(\mathcal{K}))$ of UD j is one if: 1) $j \in \mathcal{K} \cap \mathcal{N}_w$, 2) $j \in \mathcal{T} \cap \mathcal{N}_w$, 3) $j \in \mathcal{S} \cap \mathcal{N}_w$, and 4) $j \in \mathcal{N}_w \backslash \mathtt{u}_i(\mathtt{p}_i)$. Therefore, the expected overall decoding delay can be written as

$$\mathbb{E}[D(\mathcal{K}, \mathtt{p}(\mathcal{K}))] = \sum_{i\in\mathcal{N}} d_i(\mathcal{K}, \mathtt{p}(\mathcal{K})) = |\mathcal{K} \cap \mathcal{N}_w| + |\mathcal{T} \cap \mathcal{N}_w| + |\mathcal{S} \cap \mathcal{N}_w| + \sum_{i\in\mathcal{K}} \left(\sum_{j \in \mathcal{N}_w \backslash \mathtt{u}_i(\mathtt{p}_i^*)} 1 - \epsilon_{ij} \right). \tag{1}$$

Using (1), the decoding delay minimization problem can be formulated as a joint optimization over the set of transmitting UDs and their corresponding packet combinations, which can be written as follows

$$\min_{\mathcal{K}\in\mathcal{P}(\mathcal{N}), \mathtt{p}_i(\mathcal{K}) \in \mathcal{P}(\mathcal{H}_i)} |\mathcal{K} \cap \mathcal{N}_w| + |\mathcal{T} \cap \mathcal{N}_w| + |\mathcal{S} \cap \mathcal{N}_w| + \sum_{i\in\mathcal{K}} \left(\sum_{j \in \mathcal{N}_w \backslash \mathtt{u}_i(\mathtt{p}_i^*)} 1 - \epsilon_{ij} \right). \tag{2}$$

Consequently, the decoding delay minimization problem in (2) can be formulated as follows [9]

$$\mathcal{P}_1 : \max_{\mathcal{K}\in\mathcal{P}(\mathcal{N})} -|\mathcal{K} \cap \mathcal{N}_w| - |\mathcal{T} \cap \mathcal{N}_w| - |\mathcal{S} \cap \mathcal{N}_w| + \sum_{i\in\mathcal{K}} \left(\sum_{j \in \mathtt{u}_i(\mathtt{p}_i^*)} 1 - \epsilon_{ij} \right), \tag{3a}$$

$$\text{s. t.} \quad \mathtt{p}_\mathtt{i}^* = \arg \max_{\mathtt{p}_\mathtt{i} \in \mathcal{P}(\mathcal{H}_\mathtt{i})} \left(\sum_{\mathtt{j} \in \mathtt{u}_\mathtt{i}(\mathtt{p}_\mathtt{i})} 1 - \epsilon_{\mathtt{ij}} \right), \forall \mathtt{i} \in \mathcal{K}. \tag{3b}$$

Unfortunately, the outer optimization problem (3a) and the inner optimization problem (3b) of ($\mathcal{P}_1$) are inter-dependent and cannot be solved separately. Solving ($\mathcal{P}_1$) optimally using an exhaustive search is intractable [11]. Inspired by [9], this work develops a low complexity method that achieves the optimal solution to the optimization problem ($\mathcal{P}_1$).

The local IDNC graph is constructed for each UD i in the network and denoted by $\mathcal{G}_i(\mathcal{V}_i, \mathcal{E}_i)$ wherein $\mathcal{V}_i$ and $\mathcal{E}_i$ are the set of vertices and edges, respectively. Since each UD holds a subset of packets $\mathcal{H}_i$ and can serve UDs in its coverage zone $\mathcal{Z}_i$, a vertex $v_{jp} \in \mathcal{V}_i$ is generated for each packet $p \in (\mathcal{W}_j \cap \mathcal{H}_i), \forall j \in \mathcal{Z}_i$. An edge in $\mathcal{E}_i$ that connects vertices v_{jp} and $v_{j'p'}$ is created if one of the following conditions holds:

- $p = p' \Rightarrow$ Packet p is requested by both UDs j and j'.
- $p \in \mathcal{H}_{j'}$ and $p' \in \mathcal{H}_j \Rightarrow$ The packet combination $p \oplus p'$ is instantly decodable for both UDs j and j'.

III. Clustering-Based Approach

A. Cluster Generation

Essentially, two UDs can transmit simultaneously if their coverage zones are mutually disjoint. This is to avoid any conflict

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCOMM.2021.3069941, IEEE Communications Letters

transmissions between the transmitting UDs [9]. Therefore, [9] defines the cooperation graph, which is designed by generating a vertex v for each UD in the network. Two vertices v_i and v_j are connected by an edge if their coverage zones are disjoint, i.e., $\mathcal{Z}_i \cap \mathcal{Z}_j = \emptyset$. The main drawback of the cooperation graph is that it does not generate all the possible combinations of transmitting UDs. Consequently, the full cooperation graph is designed to preserve all the benefits of the cooperation graph while allowing all the possible combinations of transmitting UDs (i.e., all the possible clusters). Therefore, the full cooperation graph is designed by generating a vertex v for each cluster $C \in \mathbf{C}$ where $\mathbf{C}$ denotes all the possible clusters which can be expressed as $\mathbf{C} = \{C \in \mathcal{P}(\mathcal{N}) \ | \mathcal{Z}_k \cap \mathcal{Z}^t(C \backslash k) \neq \emptyset, \forall k \in C\}$, where $\mathcal{Z}^t(C)$ is the union of all coverage areas of UDs in C. Two clusters $C \neq C'$ interfere if their total coverage areas overlap, i.e., $\mathcal{Z}^t(C) \cap \mathcal{Z}^t(C') \neq \emptyset$. In the full cooperation graph, two vertices are connected if the total coverage areas of their corresponding clusters do not interfere.

Lemma 1. *For any particular set of transmitting UDs $\mathcal{K}$, the corresponding clustering $\Phi \in \mathcal{P}(\mathbf{C})$ satisfies the following constraints*

$$\bigoplus_{C \in \Phi} C = \mathcal{K} \tag{4a}$$

$$\mathcal{Z}^t(C) \cap \mathcal{Z}^t(C') = \emptyset, \forall C \neq C' \in \Phi, \tag{4b}$$

$$\mathcal{Z}_k \cap \mathcal{Z}^t(C \backslash k) \neq \emptyset, \forall k \in C. \tag{4c}$$

Proof. To prove that Φ associated with $\mathcal{K}$ satisfies (4a)-(4c), a sequential clustering method is provided as follows. First, we generate a cluster C by picking up an arbitrary UD k from the set $\mathcal{K}$. Afterwards, we repetitively add all remaining UDs from $\mathcal{K}$ that are interfering with C to it, until there are no UDs remaining in $\mathcal{K}$ that interfere with C. Consequently, constraint (4c) holds. Notably, the remaining UDs in $\mathcal{K}$ are not interfering with C. Second, we generate a second cluster C' by picking up one of the remaining transmitting UDs in $\mathcal{K}$ and all the interfering UDs added to it. It is easy to see that all UDs in C are not interfering with the UDs in C', and accordingly, C and C' are not interfering. By extension, clustering Φ satisfies (4b). We repeat the construction process until $\mathcal{K} = \emptyset$ which ensures that constraint (4a) holds. This concludes that Φ satisfies (4a)-(4c) for any combination of transmitting UDs $\mathcal{K}$. In (4a), $\bigoplus$ denotes the partition symbol. □

Let Ψ denote the set of all the possible clustering Φ in the network, i.e., $\Phi \in \Psi$. Given the one-to-one mapping between the set of clusters and the set of transmitting UDs as mentioned in Lemma 1, $(\mathcal{P}_1)$ can be rewritten as follows

$$\mathcal{P}_2 : \max_{\Phi \in \Psi} -|\Phi \cap \mathcal{N}_w| - |\mathcal{T} \cap \mathcal{N}_w| - |\mathcal{S} \cap \mathcal{N}_w| + \sum_{C \in \Phi} \sum_{i \in C} \left(\sum_{j \in \mathbf{u}_i(\mathbf{p}_i^*)} 1 - \epsilon_{ij} \right), \tag{5a}$$

$$\text{s. t.} \quad \mathbf{p}_\mathbf{i}^*(\Phi) = \arg \max_{\mathbf{p}_\mathbf{i} \in \mathcal{P}(\mathcal{H}_\mathbf{i})} \left(\sum_{\mathbf{j} \in \mathbf{u}_\mathbf{i}(\mathbf{p}_\mathbf{i})} 1 - \epsilon_{\mathbf{ij}} \right). \tag{5b}$$

B. Low Complexity Graph Pruning Solution

In this subsection, we propose a low complexity, yet optimal, solution for solving the decoding delay minimization problem $(\mathcal{P}_2)$. In particular, we first exhibit the condition for generating important clusters that are beneficial towards the optimal solution. Based on this, we propose a method for generating only such clusters while simultaneously constructing the multi-layer IDNC (ML-IDNC) graph.

Let $\Phi \in \Psi$ be a clustering that represents a clique in the full cooperation graph and C be a cluster in the clustering Φ. Let $k \notin C$ be a UD such that

$$\mathcal{Z}_k \cap \mathcal{Z}^t(C) \neq \emptyset. \tag{6}$$

Now, let C' be another cluster in Φ. Note that $k \notin C', \forall C' \in \Phi$. Otherwise, since k satisfies (6) then Φ violates (4b) and hence it is not a clique. Similar to Φ, let Φ^k be a clustering in which the cluster C in Φ is replaced by the cluster $C^k = C \cup \{k\}$. Thus, the clustering Φ^k is expressed as $\Phi^k = (\Phi \backslash C) \cup \{C \cup \{k\}\}$.

To determine if all the clusters are needed to achieve the optimal solution to $(\mathcal{P}_2)$, we compute the different delays of clustering Φ and clustering Φ^k. The delay for a set of transmitting UDs $\mathcal{K}$ can be expressed as the delay induced by $C \in \Phi$ and the delays induced by $C' \in \Phi^k, C' \neq C$ as illustrated in $(\mathcal{P}_2)$. Therefore, the difference in delays between Φ and Φ^k can be expressed as

$$\begin{aligned} \Delta = {} & |\mathcal{Z}^t(C) \cap \mathcal{N}_w| - |C \cap \mathcal{N}_w| - |\mathcal{T}(C) \cap \mathcal{N}_w| \\ & + y(C) - (|\mathcal{Z}^t(C^k) \cap \mathcal{N}_w| - |C^k \cap \mathcal{N}_w| \\ & - |\mathcal{T}(C^k) \cap \mathcal{N}_w| + y(C^k)). \end{aligned} \tag{7}$$

where

$$y(C) = \sum_{i \in C} \left[\max_{\mathbf{p}_i \in \mathcal{G}_i} \left(\sum_{j \in \tau_i(\kappa_i)} (1 - \epsilon_{ij}) \right) \right]. \tag{8}$$

By performing a simple manipulation to (7), we can write $|\mathcal{Z}^t(C) \cap \mathcal{N}_w| - |\mathcal{Z}^t(C^k) \cap \mathcal{N}_w|$ as $-|(\mathcal{Z}^t(C^k) \backslash \mathcal{Z}^t(C)) \cap \mathcal{N}_w|$ and $|C^k \cap \mathcal{N}_w| - |C \cap \mathcal{N}_w|$ as $|\{k\} \cap \mathcal{N}_w|$. Finally, $|\mathcal{T}(C^k) \cap \mathcal{N}_w| - |\mathcal{T}(C) \cap \mathcal{N}_w|$ can be written as $|(\mathcal{T}(C^k) \backslash \mathcal{T}(C)) \cap \mathcal{N}_w|$. Therefore, Δ can be rewritten as follows

$$\begin{aligned} \Delta = {} & -|(\mathcal{Z}^t(C^k) \backslash \mathcal{Z}^t(C)) \cap \mathcal{N}_w| + |\{k\} \cap N_w| \\ & + |(\mathcal{T}(C^k) \backslash \mathcal{T}(C)) \cap \mathcal{N}_w| + y(C) - y(C^k). \end{aligned} \tag{9}$$

If the quantity defined in (9) is a positive number, then adding UD k to the clustering Φ^k does not add any gain becasue it brings more interference than it serves UDs. Therefore, clustering Φ^k is not important becasue Φ is a cluster with a higher weight. Based on this observation, the remaining of this subsection provides an efficient method for designing a new ML-IDNC graph while judiciously generating important clusters that are certainly contributing to the decoding delay minimization.

To design the ML-IDNC graph, we first generate a vertex v_j^0 for each UD $j \in \mathcal{N}$ in the network. Let these vertices represent the first layer of the ML-IDNC graph, i.e., $v^0 \in \mathcal{G}^0, l = 0$. Two vertices v_i^0 and v_j^0 that satisfying the non-interference condition $\mathcal{Z}^t(v_j^0) \cap \mathcal{Z}^t(v_i^0) \neq \emptyset$ are connected by an edge in the graph. Afterwards, the weight of each vertex v^0, representing the cluster C^0, is computed as follows

$$\psi(v^0) = |\mathcal{Z}^t \cap \mathcal{N}_w| - |C^0 \cap \mathcal{N}_w| - |\mathcal{T} \cap \mathcal{N}_w| + y(C^0), \tag{10}$$

where $y(C^0)$ is previously defined in (8). Afterword, we construct the second layer of the ML-IDNC graph by merging

Algorithm 1: ML-IDNC Graph Design

Require: $\mathcal{Z}_i, \forall i \in \mathcal{N}$.

- Construct $\mathcal{G}^0$ and initialize $l = 0, s =$ **true**.

while $s =$ ***true*** **do**

Set $s \longleftarrow$ **false** and initialize $\mathcal{G}^{l+1} = \emptyset$.
for ***all*** $z^l \in \mathcal{G}^l$ **do**
for ***all*** $z^0 \in \mathcal{G}^0$ **do**
if $\psi(\{v^l, z^0\}) \geq \max(\psi(v^l), \psi(z^0))$ **then**
Set $s \longleftarrow$ **true**
Generate vertex $y^{l+1} = \{v^l, z^0\}$.
Set $\mathcal{G}^{l+1} \longleftarrow \mathcal{G}^{l+1} \cup y$.
end if
end for
end for
Set $l \longleftarrow l + 1$
for ***all*** $y^l \in \mathcal{G}^l$ **do**
for ***all*** $x \in \bigcup_{i=0}^{k} \mathcal{G}^k$ **do**
if $\mathcal{Z}^t(x) \cap \mathcal{Z}(y^l) = \emptyset$ **then**
Connect edge between y^l and x.
Set $\mathcal{G}^{l+1} \longleftarrow \mathcal{G}^{l+1} \cup y$.
end if
end for
end for
end while

- Set $\mathcal{G} = \bigcup_{i=0}^{l} \mathcal{G}^l$.

the clusters of the previous layer with the cluster of the first layer as long as the cooperation of the newly merged clusters is beneficial. Specifically, for each pair of two clusters v_j^0 and v_i^0 that are not connected, two scenarios can be occurred:

- $\psi(\{v_i^0, v_j^0\}) \geq \max(\psi(v_i^0), \psi(v_j^0))$: In this scenario, we generate a vertex v_{ij}^1 representing the cluster $C^1 = \{i, j\}$.
- $\psi(\{v_i^0, v_j^0\}) < \max(\psi(v_i^0), \psi(v_j^0))$: In this scenario, the cooperation is not beneficial. Thus, we do not generate the cluster C^1.

Then, the newly created vertices are connected with the other vertices if they satisfy (4b). We repeat this process for all layers of the ML-IDNC cooperation graph. For instance, for layer $l + 1$, we combine the vertices v_i^0 of the first layer ($l = 0$) with the vertices v_j^l of the previous layer ($l = 1$) if they are not connected and calculate the weight of $\psi(\{v_i^0, v_j^l\})$. Based on the weigh of the combined vertices, we decide to either generate or avoid generating that cluster. The steps of the algorithm are summarized in Algorithm 1.

Recall that a clique in undirected graphs is a set of vertices that are connected to each other, i.e., pair-wise adjacent vertices. Consequently, each clustering (a set of connected disjoint clusters) represents a clique (a set of connected vertices). Therefore, the problem of minimizing the decoding delay in ($\mathcal{P}_2$) is equivalently represented by the maximum weigh clique (MWC) problem in the ML-IDNC graph, where the weight $\psi(v)$ of each vertex v that represents the cluster C is given by

$$\psi(v) = |\mathcal{Z}^t(C) \cap \mathcal{N}_w| - |C \cap \mathcal{N}_w| - |\mathcal{T}(C) \cap \mathcal{N}_w| + \sum_{i \in C} \left(\sum_{j \in \mathtt{u}_i(\mathtt{p}_i^*)} 1 - \epsilon_{ij} \right). \quad (11)$$

Using the ML-IDNC graph in Algorithm 1, the following theorem proves the optimality of our solution to ($\mathcal{P}_2$).

Theorem 1. *The designed ML-IDNC graph generated by Algorithm 1 provides the optimal solution to ($\mathcal{P}_2$).*

Proof. First, we need to show that all the vertices in the MWC that are generated in the resulting ML-IDNC graph from Algorithm 1 are in the full cooperation graph generated in [9]. This is due to the fact that the ML-IDNC graph contains only important vertices, and thus it is a sub-graph of the full cooperation graph constructed in [9].

Let $\mathcal{C}^* = \{C_1^*, \cdots, C_{|\mathcal{C}^*|}^*\}$ be the MWC in the full cooperation graph which consists of all clusters $C_k^*, \forall\ 1 \leq k \leq |\mathcal{C}^*|$. Now, let $C_k^* = \{c_1^k, \cdots, c_{|C_k^*|}^k\} \in \mathcal{C}^*$ be a cluster where its associated UDs have been arranged as follows

$$\mathcal{Z}_{c_i^k} \cap \mathcal{Z}^T(\{c_1^k, \cdots, c_{i-1}^k\}) \neq \phi, \forall\ 1 \leq i \leq |C_k^*|. \quad (12)$$

By induction, we can prove that all the clusters C_k^* that are uniquely represented by vertices $v_i = \{c_1^k, \cdots, c_i^k\}, \forall\ 1 \leq i \leq |C_k^*|$ are generated by Algorithm 1. For $i = 1$, the UD c_1^k is included in the graph by construction. Consider that cluster v_{i-1} is included. Using (7), (9), UD c_i^k can be included in the cluster if it provides a gain to that cluster. However, this result holds only for clusters that satisfy (6). We generalize this result for only the clusters in the MWC becasue all the associated clusters in the MWC satisfy (4b), (4c).

Consider that $\psi(v_i) \leq \psi(v_{i-1})$, then we have $\psi(C_k^*) \leq \psi(C_k^* \backslash \{c_i^k\})$ and the cluster $C_k^* \backslash \{c_i^k\}$ is jointly combinable with all the clusters $C_i^*, \forall 1 \leq i \neq k \leq |\mathcal{C}^*|$. Hence, $\mathcal{C}^* = \{C_1^*, \cdots, C_{k-1}^*, C_k^* \backslash \{c_i^k\}, C_{k+1}^*, \cdots, C_{|\mathcal{C}^*|}^*\}$ offers a higher weight. Since $\mathcal{C}^*$ is the MWC, then we conclude that $\psi(v_i) \geq \psi(v_{i-1})$. Similar proof goes if $\psi(v_i) \leq \psi(c_i^k)$ in which case $\mathcal{C}^* = \{C_1^*, \cdots, C_{k-1}^*, C_k^* \backslash \{v_{i-1}\}, C_{k+1}^*, \cdots, C_{|\mathcal{C}^*|}^*\}$ offers a higher weight. In other words, we have

$$\psi(v_i) \geq \max(\psi(v_{i-1}), \psi(z_i^k)). \quad (13)$$

This concludes that cluster v_i needs to be generated. Therefore, all the clusters in the MWC are generated in the ML-IDNC graph resulting from Algorithm 1. It is worth mentioning that all the edge connections between the vertices are the same in both graphs. Therefore, the MWC in the full cooperation graph in [9] is the same as the one constructed by Algorithm 1. □

The computational complexity of generating all clusters representing all vertices in the full cooperation graph is $O\binom{|\mathcal{P}(\mathcal{N})|}{K}$. Then, connecting all these vertices requires a complexity of $O\binom{|\mathcal{P}(\mathcal{N})|}{K}^2$. Therefore, the overall computational complexity of the scheme in [9] is $O\binom{|\mathcal{P}(\mathcal{N})|}{K}^2$. In contrast, the overall complexity of our proposed solution is $O\left(L(2N^2)\right)$, where L is the number of layers in the ML-IDNC graph.

IV. Numerical Results

We consider a partially connected D2D network where the UDs are distributed randomly within a hexagonal cell of radius 500m. Every UD is connected to some other UDs within its coverage zone based on the connectivity index $\mathtt{C}$ that is defined as the ratio of the average number of neighboring UDs to the total number of UDs N. The considered model is shown

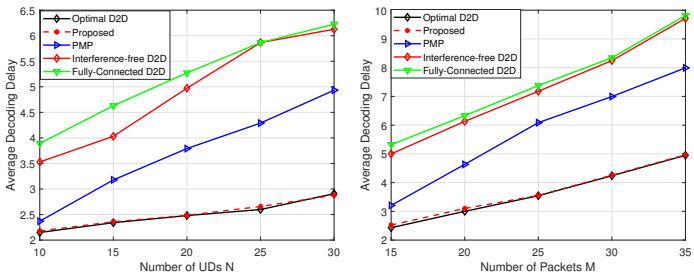

Fig. 2: Average decoding delay versus: i) number of UDs N for a network composed of $M = 15$; ii) number of packets M for a network composed of $N = 20$.

in Fig. 1 in the system model section. The simulation setting in this paper follows the setup studied in [8], [9]. Each UD has already downloaded some packets and requests some other packets from the set $\mathcal{M}$. These initial *Has* and *Wants* sets $\mathcal{H}_i$ and $\mathcal{W}_i, \forall i \in \mathcal{N}$ of UDs are arbitrarily generated based on the erasure probabilities. The D2D links are more reliable than the BS-UD communications, and accordingly, we assume that the UD-to-UD erasure probability ϵ is half the BS-to-UD erasure q, i.e., $\epsilon = 0.5q$ [8], [9]. In the simulated figures, we set $\mathtt{C} = 0.4$, $\epsilon = 0.25$, and $q = 0.5$. We executed all the simulated schemes until all the UDs receive all their requested packets. In the simulations, the decoding delay is calculated over a certain number of iterations, and the average value is presented. For the sake of performance comparison, we simulate the following schemes: 1) optimal D2D and interference-free D2D that were proposed in [9] in Sections VI, V, respectively; 2) PMP scheme where the BS is responsible for the transmissions [7]; and 3) fully-connected D2D that was proposed in [10].

From Fig. 2, we can see that the decoding delay performance of the proposed solution outperforms the decoding delay performances of PMP and fully connected D2D schemes for all network configurations. In particular, the fully-connected D2D system selects a single UD for transmission at each transmission slot. Although the BS in the PMP scheme can combine a set of packets in one IDNC file and target a set of UDs, the PMP scheme considers only one transmission and sacrifices the potential of the simultaneous transmissions and cooperation among UDs. The proposed solution balances between the aforementioned aspects by considering simultaneous transmissions from a set of cooperative transmitting UDs. In particular, the proposed scheme judiciously selects the potential UDs clustering and optimizes the packet selection process for each transmitting UD in each cluster. This utilizes all the benefits of the cooperation between the transmitting UDs. Compared to the optimal D2D scheme, our proposed solution offers the same decoding delay performance.

Finally, we assess the complexity of our proposed scheme as a function of the execution time. In setup 1, we consider a network composed of 10 UDs, 10 packets, $\epsilon = 0.25$, $q = 0.5$, and $\mathtt{C} = 0.4$. In setup 2, we consider a network composed of 15 UDs, 20 packets, $\epsilon = 0.25$, $q = 0.5$, and $\mathtt{C} = 0.4$. It can clearly be seen from Table I that the proposed scheme significantly reduces the execution time as compared to the benchmark scheme in [9] for both setups. This is due to the fact that the scheme in [9] generates all the possible combinations of the UDs in the network, which increases significantly with the number of UDs. This can be seen from Table I that when we change the number of UDs from 10 to 15, the execution time of the optimal D2D increases from 18 seconds to 257 seconds. In contrast, our proposed scheme has low execution times in both setups becasue it efficiently generates only important clusters that are minimizing the decoding delay. Therefore, our scheme provides a more effective way to use IDNC and D2D communications, both from complexity and performance perspectives. As a result, it is more suitable to be implemented in massive D2D networks.

TABLE I: Average execution times of the different schemes

Solution	Time(s)- Setup 1	Time(s)- Setup 2
Optimal D2D	18.1463	257.7268
Proposed	0.1681	0.9291
Interference-free D2D	0.0524	0.0828
Fully-connected D2D	0.0622	0.0933
PMP	0.0621	0.0839

V. Conclusion

This letter proposed a low complexity, yet optimal, NC scheme that minimizes the decoding delay in IDNC-enabled partially-connected D2D networks. Unlike the previous solution that requires generating the total number of clusters, a low-complexity solution was developed over a potential set of clusters. The developed optimization algorithm is sequentially eliminating the clusters that are certainly not minimizing the decoding delay while designing the ML-IDNC graph. Our numerical results illustrated that our proposed solution offers the same decoding delay performance similar to the benchmark solution with a significant reduction of the computational complexity.

References

[1] Cisco. (2019). *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022 White Paper.*

[2] D. Traskov, M. Medard, P. Sadeghi, and R. Koetter, "Joint scheduling and instantaneously decodable network coding," in *Proc. of IEEE Global Telec. Conf. (GLOBECOM' 2009),* Honolulu, Hawaii, USA, Nov. 2009, pp. 1–6.

[3] X. Li, C.-C. Wang, and X. Lin, "On the capacity of immediately decodable coding schemes for wireless stored-video broadcast with hard deadline constraints," *IEEE Jour. on Selected Areas in Commu.,* vol. 29, no. 5, pp. 1094-1105, May 2011.

[4] A. Douik *et al.*, "Completion time reduction in instantly decodable network coding through decoding delay control," in *Proc. of IEEE Global Telec. Conf. (GLOBECOM' 2014),* Austin, Texas, USA, Dec. 2014, pp. 5008–5013.

[5] A. Douik *et al.*, "Decoding delay-controlled completion time reduction in instantly decodable network coding," *IEEE Trans. on Veh. Tech.,* vol. 66, no. 3, pp. 2756–2770, Mar. 2017.

[6] A. Douik, M. S. Al-Abiad and M. J. Hossain, "An improved weight design for unwanted packets in multicast instantly decodable network coding," in *IEEE Comm. Letters,* vol. 23, no. 11, pp. 2122-2125, Nov. 2019,

[7] S. Sorour and S. Valaee, "Minimum broadcast decoding delay for generalized instantly decodable network coding," in *Proc. of IEEE Global Telec. Conf. (GLOBECOM' 2010),* Miami, Florida, USA, Dec. 2010, pp. 1–5.

[8] M. S. Al-Abiad, A. Douik, and M. J. Hossain, "Coalition formation game for cooperative content delivery in network coding assisted D2D communications," in *IEEE Access,* vol. 8, pp. 158152-158168, 2020.

[9] A. Douik *et al.*, "Delay reduction in multi-hop device-to-device communication using network coding," in *IEEE Trans. on Wireless Commu.,* vol. 17, no. 10, pp. 7040-7053, Oct. 2018.

[10] N. Aboutorab *et al.*, "Instantly decodable network coding for delay reduction in cooperative data exchange systems," in *Proc. of IEEE Int. Symp. on Inf. Theory (ISIT' 2013),* Istanbul, Turkey, July 2013, pp. 3095–3099.

[11] K. Yamaguchi and S. Masuda, "A new exact algorithm for the maximum weight clique problem," in *Computers and Commun. 23rd Intern. Conf.on Circuits/Systems (ITC-CSCC'08),* 2008.